UDC 004.855.6
IRSTI 20.19.19

https://doi.org/10.55452/1998-6688-2025-22-2-141-154

[1]*Zhangbyrbay Zh.,
Master's student, ORCID ID: 0009-0003-5568-6142,
*e-mail: z_zhangbyrbay@kbtu.kz
[2]Akhmetov I.,
PhD, professor, ORCID ID: 0000-0002-3221-9352,
e-mail: i.akhmetov@ipic.kz
[1]Pak A.,
PhD, professor, ORCID ID: 0000-0002-8685-9355,
e-mail: a.pak@kbtu.kz
[1]Jaxylykova A.,
PhD student, ORCID ID: 0000-0003-0422-7432,
e-mail: a.jaxylykova@kbtu.kz
[3]Komada P.
PhD, professor, ORCID ID: 0000-0002-9032-9285,
e-mail: p.komada@pollub.pl

[1]Kazakh-British Technical University, Almaty, Kazakhstan
[2]Institute of Information and Computational Technologies, Almaty, Kazakhstan
[3]Lublin University of Technology, Lublin, Poland

ADAPTATION OF TEXT GENERATION STYLE TO A SPECIFIC AUDIENCE OR CONTENT

Abstract

Adaptation of text generation style to specific audiences or content can be achieved without costly fine-tuning. We freeze model weights and instead (i) search eight decoder hyperparameters with Bayesian optimization and (ii) prepend a one-line style cue that modulates readability. Experiments on five mathematical question-answering benchmarks (AQUA-RAT, MathQA, GSM8K, MAWPS, SVAMP) with three 8–14 B-parameter checkpoints (LLaMA-3.1-8B, DeepSeek-Qwen-8B/14B) show that 50-trial Optuna searches raise exact-match accuracy by up to 36 percentage points and close 5–10 points of the gap to 30–70 B fine-tuned baselines. The same settings transfer across tasks with under 2-point loss. Adding the child-friendly header leaves accuracy virtually unchanged while halving the Flesch–Kincaid grade level and shortening reasoning traces. All experiments fit within a few GPU-hours on a single A100, making the method practical for resource-constrained deployments. The study demonstrates that careful decoder control combined with micro-prompts delivers numerical correctness and audience-appropriate exposition without additional training or tuning time.

Keywords: decoder optimization, style adaptation; readability, large language models, mathematical question answering, Bayesian hyper-parameter search, Flesch–Kincaid score.

Introduction

The quick transition of large language models (LLMs) from open-ended chat to specialized applications in legal drafting, biomedical summarization, and quantitative reasoning domains has occurred rapidly. Open checkpoints typically do not fulfill two essential requirements: numerical accuracy and an appropriate explanation for the audience. Large models between 70–540 B parameters provide scale-based solutions to the first criterion in, but prompt engineering deals with the second criterion as in [1–3]. The implementation of these solutions requires substantial effort through either

GPU-hour expenses or manual prompt development, which leaves practitioners who use mid-scale weights (6–14 B) and automated pipelines without an effective solution [4].

New research demonstrates that fine-tuned decoder management enables hidden capabilities without modifying model weights. The use of temperature adjustment together with top- k sampling techniques helps prevent the "neural text degeneration" effect, and beam-diversity heuristics enhance factual accuracy in translation tasks [5–7]. Most existing systematic searches remain rare in literature because numerous studies continue to depend on grid heuristics despite proven superiority of Bayesian optimization over grid and random kernels, as shown in [8–10]. Optuna and BOHB reduce the expense of systematic searches by cutting short trials that perform poorly, which results in training-free performance improvements [11, 12].

The prompting techniques chain-of-thought (CoT) [2] and self-consistency [13] show how a single sentence can change the answer structure. CoT frequently results in longer outputs that elevate the complexity of text beyond basic reading levels, thus restricting its effectiveness for people who are young or have limited literacy skills. The field of readability adaptation has not received enough research attention for math QA, especially since word problems exist in K–12 curricula [14–15].

This paper examines the boundary of achievable performance gains in mid-scale mathematical benchmarks through the use of decoder parameter adjustments with lightweight style cues. Empirical setup. Three publicly available 8–14B checkpoints–LLaMA-3.1-8B, DeepSeek-Qwen-8B, and DeepSeek-Qwen-14B–are utilized for evaluation. The analysis is conducted across five benchmark datasets: AQUA-RAT (multi-choice algebra) [16], MATHQA (operation-based mix) [13], GSM8K (grade-school multi-step) [17], MAWPS (single-step repository) [18], and SVAMP (concept-transfer traps) [15]. An Optuna-TPE search uses eight decoding knobs-temperature, top-k/p, repetition and length penalties, beam width, and token limits-for each dataset across 50 trials.

The selected best settings remain fixed while the model receives two separate headers that are mutually exclusive:

(i) "explain so a 12-year-old can follow" and (ii) "provide a formal derivation using inline LATEX." Accuracy is measured exactly, while readability is quantified via the Flesch–Kincaid

Grade Level (FKGL) index.[1]

Key findings (preview). (1) The first finding shows that GSM8K reaches 83.33% accuracy after decoder optimization on LLaMA-3.1, 82.3% accuracy on Qwen-8B, and 90.7 % accuracy on Qwen-14B without gradient updates, increasing the score by 17 points. (2) Multiple-step datasets use a common optimization approach, which consists of setting temperatures at 1.5–1.9 and top-k values between 150–180 while implementing strong repetition penalties ($\rho \approx 2$) and choosing 3–4 beams. A single style header maintains both accuracy at 2 pp while decreasing the FKGL score from 7.3 to 3.6 for child-friendly text and extends chain-of-thought by 25% for formal proofs. The effects observed in this study demonstrate consistency across different datasets. The application of GSM8K-tuned decoding to MATHQA results in a 55.4% performance score, which is 1.9 pp lower than the score obtained through MATHQA-specific tuning, thus supporting the cross-task robustness findings of PAL [19].

Contributions

1) This study establishes the first comprehensive analysis that compares Bayesian optimizer performance on three mid-scale models when applied to five math QA datasets while achieving better results without requiring retraining.

2) The research measures how single-line audience signals affect reading comprehension alongside reasoning complexity without compromising numerical accuracy, which enlarges usability research on CoT [2] to include K–12 environments.

3) The findings show that by adjusting the decoder while using micro-prompts, the performance approaches that of 30–70 B baselines, thus providing a useful framework for limited-resource deployments [20, 21].

Related work

The complete solution requires exploration of three distinct bodies of literature: hyper-parameter optimization methods, token-decoding approaches for large language models, and specialized

techniques for mathematical problem solving. Each topic is elaborated on below, with clarification provided on how the present research relates to these lines of work.

A. Hyperparameter Optimization (HPO): Resource allocation in HPO has shifted from exhaustive grid search toward advanced bandit-style methods [22]. Early work showed that plain random search can outperform grid search in high-dimensional spaces [8], spurring probabilistic techniques such as Gaussian process Bayesian optimization [9], sequential model-based configuration (SMAC) [10], and tree structured Parzen estimators (TPE) [23]. Parallelism and early stopping later appeared: Hyperband [24] allocates more budget to promising configurations, while BOHB [12] marries Hyperband's pruning with TPE's density estimation.

Frameworks including Hyperopt, Ray Tune, Vizier, and Optuna now support distributed, asynchronous search with user-defined pruning [11]. Most studies, however, still target training hyper-parameters (learning rate, batch size); systematic optimisation of inference knobs is uncommon. Previous approaches such as adjusting translator beam width or speech recognition language model weights typically involve only two or three variables. In contrast, the present study investigates an eight-dimensional inference space across three mid-scale LLMs, demonstrating that a 50-trial budget is sufficient to eliminate double-digit accuracy gaps. This result aligns with the cost-efficiency principles underlying methods such as Hyperband and BOHB.

[1]FKGL = 0.39 (W/S) + 11.8 (Syll/W) − 15.59, where W , S, and Syll denote word, sentence, and syllable counts.

B. LLM Token-Decoding Strategies: An autoregressive decoder maps a softmax distribution over thousands of tokens to a single next token; the algorithmic choice critically affects factuality, diversity, and hallucination. Greedy and beam search from the machine translation era [6] maximise likelihood but yield length bias and dull repetitions. Top-k sampling [25] trims the probability tail, and nucleus (top-p) sampling [5] dynamically sizes the candidate set. Repetition "spring-back" methods like the CTRL penalty [26] and unlike- lihood training [27] suppress loops, though they are usually applied during training. Shi et al. [28] benchmarked dozens of decoders on summarisation and story-generation tasks but omitted single-answer tasks such as math QA.

Most decoding papers report only one or two manually tuned settings. Bayesian search is applied to six interdependent parameters-temperature, top-k/top-p, repetition and length penalties, beam width, and token limits across five quantitative datasets. This approach addresses the gap identified by Holtzman et al. in the context of neural text degeneration [5]. Our results confirm that multi-step reasoning prefers hotter, broader sampling with strong repetition control, whereas single-step arithmetic is best served by cooler, deterministic decoding.

C. Mathematical Reasoning with LLMs:

Template-matching systems of the 1980s were early benchmarks for language-based reasoning, but modern datasets-MAWPS [18], SVAMP [15], GSM8K [17] expose failures in both symbolic and neural approaches. Prompt engineering produced a breakthrough: chain-of-thought (CoT) prompts [2] lift GPT-3's GSM8K accuracy above 55%, and self-consistency [13] adds a further 10 pp by sampling multiple reasoning paths. Yet Wei et al. still report only 18% for the 6 B variant [2]. Large-scale fine-tuning, as in Minerva-62 B [20] and the MATH benchmark [21], improves accuracy but consumes megawatt-hours of compute.

Our work is orthogonal: like Huang et al.'s verifier study [17], model weights are kept fixed, with all operations performed exclusively on the decoder. Proper sampling raises GSM8K to 83.3% on LLaMA-3.1-8B and surpasses several 30–40 B baselines; style cues further vary readability from FKGL 3.6 to 7.3 without harming accuracy, establishing the first systematic link between readability metrics and decoder hyper-parameters in math QA.

D. Gaps Addressed in This Paper: Existing research either tunes training parameters with heavy compute, hand-picks a few inference knobs, or extends CoT to enhance reasoning without audience adaptation. All three axes are unified through: (i) cost-efficient Bayesian optimization of eight inference parameters, (ii) evaluation across five mathematical benchmarks and three mid-scale

models, and (iii) the use of single-line style prompts that balance readability and accuracy. This combination augments the HPO toolkit [11, 12], deepens the decoding survey of Shi et al. [28], and adds training-free functionality to CoT-centric solvers [2, 13, 20].

Materials and Methods

Our workflow is a two–stage, purely inference–time pipeline. Stage 1 performs Bayesian hyper-parameter search on fixed validation sub samples; Stage 2 re-generates those same rows with a one-line style cue, enabling a direct accuracy-versus-readability comparison.

A. Stage 1 – Bayesian Search on Validation Sub Samples Sampling protocol: Full passes over some benchmarks exceeded practical runtime (e.g. 7 k rows GSM8K). Deterministic subsets are drawn accordingly:

- 300 rows each for GSM8K, MATHQA, and AQUA- RAT;
- entire MAWPS (1 084 rows) and SVAMP (695 rows).

Search variables and bounds: Table I lists the eight decoder knobs and their task-agnostic ranges. Continuous variables follow uniform priors; discrete ones are sampled uniformly.

Table 1 – Hyper-parameter search space

Variable	Range
Temperature T	0.2 . . . 2.0
Top-k	5 . . . 400
Top-p	0.5 . . . 1.0
Repetition penalty ρ	1.0 . . . 2.0
Length penalty λ	0.5 . . . 2.0

Optimizer: Optuna–TPE [11], [23] is employed with n_trials set to 50 for each dataset. Trials whose partial accuracy falls below the running 25-th percentile after 30% of their token budget are pruned (Hyperband heuristic [24]). Objective = exact-match accuracy on the subset; a single wrong digit yields 0 for that item.

B.Canonical Prompt Template

Decoding is driven by a 4-shot, chain-of-thought pattern shared across all datasets:

Question:
<question_0>
Result:
<answer_0>
1FKGL = 0.39 (W/S) + 11.8 (Syll/W) − 15.59, where W , S, and Syll denote word, sentence, and syllable counts.

Question:
<question_1>
Result:
<answer_1>

Question:
<question_2>
Result:
<answer_2>

Question:

<question_3>
Result:
<answer_3>

Question:
<question_i>
Result:

The final <question_i> is the row being solved. The model must end its answer with the line Result: <numeric>, enabling exact string comparison.

The final <question_i> is the row being solved. The model must end its answer with the line
Result: <numeric>, enabling exact string comparison.

C. Stage 2 – Readability-Oriented Text Adaptation

After Stage 1 has produced the best decoder configuration, the hyperparameters are frozen, and two stylistic variants are generated for each item in the validation subsample. Baseline / technical: exactly the same prompt used during optimization (no additional header).

1) Child-friendly: the identical prompt but preceded by the single <system> instruction "Explain step by step so a twelve year old can follow."

No further hyperparameter search is performed; the comparison isolates the influence of a one-line audience cue. Three metrics are recorded for each item:

- Exact-match accuracy - identical criterion to Stage 1.
- Readability – Flesch–Kincaid Grade Level (FKGL).2
- Chain-of-thought (CoT) length – token count up to, but excluding, the first digit in the final answer.

The qualitative examples and FKGL statistics reported in Section IV (Table VI) stem directly from this two variant generation procedure: the child header more than halves mean FKGL (7.34 → 3.56) while reducing accuracy by only 2 pp.

D. Model, Tokeniser, and Execution Pipeline

Checkpoints: Evaluations are conducted on LLaMA-3.1-8B, DeepSeek-Qwen-8B, and DeepSeek-Qwen-14B models. All models are loaded using 4-bit NF4 quantization implemented through bitsandbytes.

2FKGL = 0.39 (W/S) + 11.8 (Syll/W) − 15.59, where W and S are word and sentence counts.

model_id = "deepseek-ai/DeepSeek-R1-Distill-Qwen-7B" tokenizer = AutoTokenizer.from_ pretrained(model_id,

device_map="auto", token=hf_token)

model = AutoModelForCausalLM.from_pretrained(model_id, quantization_config=bnb_ config, device_map="auto",

token=hf_token)

for i in range(4, sample_size):

prompt = build_five_shot_prompt(i) # template above

inputs = tokenizer(prompt, return_tensors="pt"). to(device)

with torch.no_grad():

out = model.generate(**inputs, **best_cfg) #

Optuna result

txt = tokenizer.decode(out[0], skip_special_tokens=True)

pred = extract_numeric(txt, prompt) # string ops

Hardware and cost: All experiments are executed on a single NVIDIA A100-PCIE-40GB GPU. Runtime scales linearly with the number of Optuna trials and roughly linearly with the beam width: runs with num_beams = 4 are noticeably slower than their greedy counterparts. Even with subset evaluation, a 50-trial study per dataset remains computationally expensive and can take many GPU-hours, especially for the larger 14-billion-parameter checkpoint.

Results

A. Overview Across Three Models

Table II contrasts default decoding with the Optunatuned settings for all three checkpoints. Two trends emerge immediately:

1) Decoder tuning is model-agnostic: every dataset–model pair improves, with gains ranging from +1.6 to +40 pp.

2) Smaller models profit more: Qwen-8B sees the largest deltas (up to +35.7 pp on GSM8K), while the already- strong LLaMA gains a respectable +16.7 pp on the same slice.

B. Best Hyper-Parameter Settings

Tables III, IV, and V list the winning configurations discovered by Optuna. Although ranges were shared, the optimizer converged on markedly different regimes.

C. Cross-Model Observations

Entropy vs. scale. Smaller Qwen-8B requires hotter and wider sampling (T = 0.85 with k = 355 on GSM8K) to match the diversity naturally present in the larger checkpoints.

Beam width. Qwen-14B prefers fewer beams (often b = 2), suggesting that its internal representation already covers diverse trajectories; LLaMA gains from b = 3−4.

D. Token budget. Across all models, SVAMP needs the shortest answers (N_{max} = 33 for LLaMA), whereas AQUA-RAT and MATHQA push towards the 300–380 ceiling, aligning with their verbose rationales.

E. Runtime Impact of Beam Width

Although exact GPU time varies by model, increasing num_beams from 1 to 4 roughly doubles decoding latency at fixed hyper-parameters. Hence, practitioners should weigh the +4–8 pp accuracy gain against a 2× cost multiplier.

F. Qualitative Readability Study

To illustrate how the child-friendly header reshapes prose, Table VI shows verbatim outputs for five randomly chosen GSM8K items-once with the default tuned prompt and once with the child header. The Flesch–Kincaid Grade Level (FKGL) is subsequently computed for each answer.

Across these five examples, the child header cuts the average FKGL from 7.34 (middle–school level) to 3.56 (early elementary) while retaining the exact numeric answer in every case. Notably, item 4 remains relatively complex because of unit conversions, indicating that some problems are intrinsically harder to simplify.

G. Summary

Decoder-level Bayesian optimization delivers sizeable, model-agnostic accuracy gains: up to +40 pp for the smaller Qwen-8B and a consistent +15–17 pp for LLaMA-3.1-8B. Hyper-parameter optima cluster by task complexity (hot, wide sampling for multi-step algebra; cool, narrow decoding for single-step arithmetic) and by model scale (larger checkpoints require fewer beams and lower entropy). Crucially, our second- stage text-adaptation experiment shows that adding a single audience header can halve the FKGL readability score (7.34 → 3.56) or lengthen formal derivations by 25 % while preserving at least 95 % of the tuned accuracy. Taken together, the two stages push mid-scale models to within striking distance of 30–70 B fine-tuned baselines-at a fraction of the computational and prompting cost, and with the added benefit of audience-specific presentation.

Results and Discussion

The paper combines empirical results in four directions:

(i) decoder patterns that maintain consistency across multiple datasets, (ii) how these patterns shift with model scale, (iii) the impact of a one-line child cue on readability and accuracy, and (iv) practical implications for real-world deployment.

A. Decoder Patterns Across Datasets: Tables III–V present configurations that demonstrate a definitive distinction: for GSM8K, AQUA-RAT, and MATHQA the optimizer con- verges on T ≈ 1.5−2.0, k ≳ 150, strong repetition penalties ь(ρ ≈ 1.9), and 3–4 beams, whereas single-step or "trap" corpora (SVAMP) maintain high temperature values but re- duce the candidate set to k = 20−30 and relax ρ. Deeper reasoning thus benefits from wide exploration plus strong loop suppression, while adversarial distractors require tight focus to prevent semantic drift.

Table 2 – Baseline (Default) vs. tuned (Optimized) accuracy on each validation sub sample. Δ = absolute improvement in percentage points. A dash (-) indicates that the model was not evaluated on that corpus owing to GPU-time constraints

	LLaMA-3.1-8B			Qwen-8B			Qwen-14B		
Dataset	Default	Optim.	Δ	Default	Optim.	Δ	Default	Optim.	Δ
GSM8K	66.7	83.3	+16.7	46.7	82.3	+35.7	77.7	90.7	+13.0
MathQA	43.7	57.3	+13.7	—	—	—	—	—	—
AQUA-RAT	40.3	57.7	+17.3	24.7	39.7	+15.0	49.0	70.0	+21.0
MAWPS	88.8	90.4	+1.6	48.6	52.3	+3.7	44.1	75.6	+31.5
SVAMP	62.8	65.7	+2.9	35.1	55.5	+20.4	43.3	70.2	+27.0

Table 3 – Best settings – LLaMA-3.1-8B

	GSM	MQA	AQUA	MWPS	SVAMP
T	1.44	0.95	1.78	0.33	1.96
k	184	148	166	283	20
p	0.68	0.98	0.74	0.50	0.51
ρ	1.99	1.79	1.90	1.73	1.03
λ	1.16	1.11	0.82	0.56	0.77

Table 4 – Best settings – QWEN-8B

	GSM	MQA	AQUA	MWPS	SVAMP
T	0.85	–	1.99	1.64	1.25
k	355	–	258	341	122
p	0.93	–	0.55	0.55	0.70
ρ	1.12	–	1.00	1.19	1.07
λ	1.96	–	0.51	1.86	1.60
Nmax	314	–	296	340	191
Nmin	3	–	19	47	17
b	3	–	3	2	2

Table 5 – Best settings – QWEN-14B

	GSM	MQA	AQUA	MWPS	SVAMP
T	0.72	–	1.13	1.66	1.66
k	388	–	281	182	388
p	0.86	–	0.65	0.99	0.70
ρ	1.95	–	1.08	1.37	1.40
λ	0.93	–	1.70	0.95	1.84
Nmax	323	–	382	341	284
Nmin	49	–	48	44	6
b	2	–	2	3	2

B. Role of Individual Hyper-Parameters: Temperature. Dropping T below 1.0 lowers accuracy on smaller GSM8K checkpoints by 6–8 pp, but the 14 B model is nearly un- affected, indicating that parameter count can substitute for entropy.

Top-k & Top-p. AQUA-RAT peaks at k = 150, p = 0.7; smaller k truncates valid algebraic phrases, larger k injects noise. SVAMP is the exception, functioning optimally with k = 20.

Repetition penalty. Raising ρ to 2.0 removes 70 % of loop errors on GSM8K but hurts SVAMP, where legitimate token repetition occurs.

Length controls. Verbose corpora reward full derivations with Nmax ≥ 360 and λ < 1; SVAMP caps output at Nmax = 33. Beam width. Three or four beams boost accuracy by 4–8 pp across all datasets except SVAMP; doubling beams roughly doubles latency, revealing a speed–quality trade-off.

C. Effect of the Child-Friendly Header: The child cue reduces mean FKGL from 7.34 to 3.56 on the five-item mini- corpus (Table VI) and cuts full-subset accuracy by only ≈ 2pp. Chain-of-thought length shortens slightly (44→38 tokens on GSM8K); the model prunes elaborate sentences but retains the numeric solution. Readability therefore appears orthogonal to correctness once a dependable decoder regime is fixed.

D. Model-Scale Effects: Qwen-8B gains the most (+35.7 pp on GSM8K); LLaMA-8B gains a consistent +16–17 pp; Qwen-14B still improves (+13 pp) but shows diminishing returns, implying decoder tuning is most valuable when model size is hardware-limited.

E. Cost and Deployment Concerns: Running 50 Optuna trials on a 300-row subset takes several GPU-hours on a single A100, yet the discovered settings transfer well: only –1.9 pp when porting GSM8K settings to MATHQA. Early pruning and subset evaluation keep the search viable for practitioners.

F. Limitations and Future Work:

- Sub sample bias: full-corpus accuracy may differ by a few points.
- Prompt scope: only one 4-shot template is tested; future work could co-optimise shot count and decoder parameters.
- Numeric parsing: string heuristics miss answers with commas or units; a regex-robust parser would improve accuracy.

Possible extensions include dynamic temperature schedules, joint prompt–decoder optimisation, and lightweight verifier functions for premise-parsing errors.

G. Take-Away: Bayesian tuning of eight decoder variables, combined with a single child-friendly signal, lets mid-scale LLMs approach 30–70 B baselines by 5–10 pp while sharply lowering readability scores an attractive trade-off for education and other resource-constrained applications.

Table 6 – Baseline VS. CHILD-friendly generations. FKGL is computed per answer; lower is simpler

ID	Baseline answer (excerpt)	Child-friendly answer (excerpt)	FKGL
1	“We start with Maria’s initial inventory of 12 ap- ples. She then transfers ownership of 5 apples to John, leaving her with 12 minus 5 = 7 apples. Subsequently, she purchases an additional tranche of 7. . . ”	“Maria has 12 apples. She gives 5 to John, so now she has 7. She buys 7 more... Maria ends with 14 apples.”	8.8 → 0.7

Continuation of table 6

2	“The rectangular garden has a length of 18 m and a width of 12 m. Its perimeter is computed as two times (18 + 12) = 60 m. Each meter of fencing costs $2.25. . . ”	“Add the two sides: 18 m + 12 m = 30 m. Double it: 30 × 2 = 60 m. Each meter costs $2.25. . . The fence costs $135.”	6.1 → 4.1
3	“Let x be the original price of the book. After a 20 % discount, the customer pays 0.8x dollars. A subsequent 8 % sales tax is imposed. 0.864x = 25.92...”	“First part: 150 km in 2 h gives 75 km/h. . . 330 ÷ 5 = 66 km/h. So the average speed is 66 km/h.”	5.1→0.9
4	“A car travels 150 km in 2 h. . . Total displacement is 330 km. . . the average speed over the trip equals 66 km/h.”	“First part: 150 km in 2 h gives 75 km/h. . . 330 ÷ 5 = 66 km/h. So the average speed is 66 km/h.”	8.9 → 8.5
5	“The sequence follows the quadratic pattern an = n2 + n. To find the eleventh term. 132.”	“The rule is n squared plus n. For n = 11. . . The eleventh number is 132.”	7.8 → 3.6
Mean	Baseline 7.34 ⇒ Child 3.56		

Conclusions

Summary of Achievements

This paper demonstrates that decoder-only optimization is a high-leverage lever for mid-scale language models.

◆ Across three checkpoints. A 50-trial Optuna search lifts LLaMA-3.1-8B by +16.7pp on GSM8K, Qwen- 8B by +35.7pp, and even the stronger Qwen-14B by +13pp without touching a single weight.

◆ Across five benchmarks. Every dataset improves: +17.3pp on AQUA-RAT, +31.5pp on MAWPS (Qwen-14B), and smaller but significant gains on SVAMP. Multi-step corpora converge on $\langle\, T = 1.5 - 2.0,\ k \gtrsim 150,\ \rho \approx 2,\ b = 3 - 4 \,\rangle$; single-step traps shrink k and relax ρ.

◆ Readability at no cost. Inserting a single child-friendly header after tuning trims mean FKGL from 7.3 to 3.6 and shortens chain-of-thought by 15% while the exact-match score drops by at most 2pp.

◆ Resource efficiency. The entire optimization runs on one A100 GPU per dataset; early pruning and 300-row sub samples keep wall-clock cost to a few hours. The discovered settings transfer: applying the GSM8K optimum to MATHQA loses only 1.9pp, allowing the search cost to be amortized across tasks.

Collectively, these results close ≈5–10pp of the gap to 30–70B fine-tuned baselines while adding a tunable readability knob compelling for education and compute-constrained deployments.

Limitations

1) Subset bias. Optimization is performed on fixed 300-row slices; full test sets may result in accuracy variations of several points.

2) Prompt invariance. Only one 4-shot template is used; different few-shot mixes might alter the optimal decoder regime.

3) Numeric parsing. Our answer extractor is string-based; comma-separated or unit-tagged numbers are discarded, slightly under-reporting true accuracy.

Where Next?

Dynamic decoders: Anneal temperature or beam width as generation unfolds, mimicking "thought hard then speak plainly" strategies.

Joint prompt–decoder search: Optimize few-shot examples and decoder knobs in a single Bayesian loop, potentially with a multi-objective (accuracy + FKGL) reward.

Verifier-in-the-loop: Plug lightweight arithmetic checkers or symbolic solvers into the decoding beam; early experiments suggest another 3–5pp may be recoverable.

Domain transfer: Evaluate the same eight-knob search on chemistry explanations, financial reasoning, or legal drafting; preliminary tests on MATHQA and AQUA-RAT show promising cross-task robustness.

Human-in-the-loop readability: Collect classroom feed- back to refine the child header, targeting specific grade levels or languages other than English.

Take-away: Inference-time Bayesian tuning, followed by a one-line style cue, is a low-cost recipe for turning mid-scale LLMs into accurate, audience-aware problem solvers no gradient steps required.

Acknowledgment

This research was funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. AP23489782)

REFERENCES

1 Brown T. et al. Language models are few-shot learners // Advances in neural information processing systems. – 2020. – Vol. 33. – P. 1877–1901.

2 Wei J. et al. Chain-of-thought prompting elicits reasoning in large language models // Advances in neural information processing systems. – 2022. – Vol. 35. – P. 24824–24837.

3 Kojima T. et al. Large language models are zero-shot reasoners // Advances in neural information processing systems. – 2022. – Vol. 35. – P. 22199–22213.

4 Touvron H. et al. Llama: Open and efficient foundation language models //arXiv preprint arXiv:2302.13971. – 2023.

5 Holtzman A. et al. The curious case of neural text degeneration // arXiv preprint arXiv:1904.09751. – 2019.

6 Wu Y. et al. Google's neural machine translation system: Bridging the gap between human and machine translation // arXiv preprint arXiv:1609.08144. – 2016.

7 Ippolito D. et al. Comparison of diverse decoding methods from conditional language models // arXiv preprint arXiv:1906.06362. – 2019.

8 Bergstra J., Bengio Y. Random search for hyper-parameter optimization // The journal of machine learning research. – 2012. – Vol. 13. – No. 1. – P. 281–305.

9 Snoek J., Larochelle H., Adams R.P. Practical bayesian optimization of machine learning algorithms // Advances in neural information processing systems. – 2012. – Vol. 25.

10 Hutter F., Hoos H. H., Leyton-Brown K. Sequential model-based optimization for general algorithm configuration // Learning and intelligent optimization: 5th international conference, LION 5, Rome, Italy, January 17–21, 2011. selected papers 5. – Springer Berlin Heidelberg, 2011. – P. 507–523.

11 Akiba T. et al. Optuna: A next-generation hyperparameter optimization framework // Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. – 2019. – P. 2623–2631.

12 Falkner S., Klein A., Hutter F. BOHB: Robust and efficient hyperparameter optimization at scale // International conference on machine learning. – PMLR, 2018. – P. 1437–1446.

13 Wang X. et al. H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models //The Eleventh International Conference on Learning Representations. – 2023. – Vol. 1.

14 Amini A. et al. Mathqa: Towards interpretable math word problem solving with operation-based formalisms // arXiv preprint arXiv:1905.13319. – 2019.

15 Patel A., Bhattamishra S., Goyal N. Are NLP models really able to solve simple math word problems? //arXiv preprint arXiv:2103.07191. – 2021.

16 Ling W. et al. Program induction by rationale generation: Learning to solve and explain algebraic word problems //arXiv preprint arXiv:1705.04146. – 2017.

17 Cobbe K. et al. Training verifiers to solve math word problems //arXiv preprint arXiv:2110.14168. – 2021.

18 Koncel-Kedziorski R. et al. MAWPS: A math word problem repository //Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies. – 2016. – P. 1152–1157.

19 Gao L. et al. Pal: Program-aided language models // International Conference on Machine Learning. – PMLR, 2023. – P. 10764–10799.

20 Lewkowycz A. et al. Solving quantitative reasoning problems with language models //Advances in Neural Information Processing Systems. – 2022. – Vol. 35. – P. 3843–3857.

21 Hendrycks D. et al. Measuring mathematical problem solving with the math dataset // arXiv preprint arXiv:2103.03874. – 2021.

22 Feurer M., Hutter F. Hyperparameter optimization. – Springer International Publishing, 2019. – P. 3–33.

23 Bergstra J., Yamins D., Cox D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures // International conference on machine learning. – PMLR, 2013. – P. 115–123.

24 Li L. et al. Hyperband: A novel bandit-based approach to hyperparameter optimization // Journal of Machine Learning Research. – 2018. – Vol. 18. – No. 185. – P. 1–52.

25 Fan A., Lewis M., Dauphin Y. Hierarchical neural story generation // arXiv preprint arXiv:1805.04833. – 2018.

26 Keskar N. S. et al. Ctrl: A conditional transformer language model for controllable generation //arXiv preprint arXiv:1909.05858. – 2019.

27 Pillutla K. et al. Mauve: Measuring the gap between neural text and human text using divergence frontiers // Advances in Neural Information Processing Systems. – 2021. – Vol. 34. – P. 4816–4828.

28 Shi C. et al. A thorough examination of decoding methods in the era of llms // arXiv preprint arXiv:2402.06925. – 2024.

REFERENCES

1 Brown T. et al. Language models are few-shot learners. Advances in neural information processing systems, 33, 1877–1901 (2020).

2 Wei J. et al. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 35, 24824–24837 (2022).

3 Kojima T. et al. Large language models are zero-shot reasoners. Advances in neural information processing systems, 35, 22199–22213 (2022).

4 Touvron H. et al. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023).

5 Holtzman A. et al. The curious case of neural text degeneration. arXiv preprint arXiv:1904.09751 (2019).

6 Wu Y. et al. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016).

7 Ippolito D. et al. Comparison of diverse decoding methods from conditional language models. arXiv preprint arXiv:1906.06362 (2019).

8 Bergstra J., Bengio Y. Random search for hyper-parameter optimization. The journal of machine learning research,13 (1), 281–305 (2012).

9 Snoek J., Larochelle H., Adams R.P. Practical bayesian optimization of machine learning algorithms. Advances in neural information processing systems, 25 (2012).

10 Hutter F., Hoos H. H., Leyton-Brown K. Sequential model-based optimization for general algorithm configuration. Learning and intelligent optimization: 5th international conference, LION 5, Rome, Italy, January 17–21, 2011, selected papers 5 (Springer Berlin Heidelberg, 2011), pp. 507–523.

11 Akiba T. et al. Optuna: A next-generation hyperparameter optimization framework. Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 2623–2631 (2019).

12 Falkner S., Klein A., Hutter F. BOHB: Robust and efficient hyperparameter optimization at scale. International conference on machine learning. PMLR, 1437–1446 (2018).

13 Wang X. et al. H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. The Eleventh International Conference on Learning Representations, 1 (2023).

14 Amini A. et al. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. arXiv preprint arXiv:1905.13319 (2019).

15 Patel A., Bhattamishra S., Goyal N. Are NLP models really able to solve simple math word problems? arXiv preprint arXiv:2103.07191 (2021).

16 Ling W. et al. Program induction by rationale generation: Learning to solve and explain algebraic word problems. arXiv preprint arXiv:1705.04146 (2017).

17 Cobbe K. et al. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168 (2021).

18 Koncel-Kedziorski R. et al. MAWPS: A math word problem repository. Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies, 1152–1157 (2016).

19 Gao L. et al. Pal: Program-aided language models. International Conference on Machine Learning (PMLR, 2023), pp. 10764–10799.

20 Lewkowycz A. et al. Solving quantitative reasoning problems with language models. Advances in Neural Information Processing Systems, 35, 3843–3857 (2022).

21 Hendrycks D. et al. Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874 (2021).

22 Feurer M., Hutter F. Hyperparameter optimization. Springer International Publishing, 2019, 3–33.

23 Bergstra J., Yamins D., Cox D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. International conference on machine learning. PMLR, 2013, 115–123.

24 Li L. et al. Hyperband: A novel bandit-based approach to hyperparameter optimization. Journal of Machine Learning Research, 18 (185), 1–52 (2018).

25 Fan A., Lewis M., Dauphin Y. Hierarchical neural story generation. arXiv preprint arXiv:1805.04833 (2018).

26 Keskar N.S. et al. Ctrl: A conditional transformer language model for controllable generation. arXiv preprint arXiv:1909.05858 (2019).

27 Pillutla K. et al. Mauve: Measuring the gap between neural text and human text using divergence frontiers. Advances in Neural Information Processing Systems, 34, 4816–4828 (2021).

28 Shi C. et al. A thorough examination of decoding methods in the era of llms. arXiv preprint arXiv:2402.06925 (2024).

[1]*Жаңбырбай Ж.,
магистрант, ORCID ID: 0009-0003-5568-6142,
*e-mail: z_zhangbyrbay@kbtu.kz
[2]Ахметов И.,
PhD, профессор, ORCID ID: 0000-0002-3221-9352,
e-mail: i.akhmetov@ipic.kz
[1]Пак А.,
PhD, профессор, ORCID ID: 0000-0002-8685-9355,
e-mail: a.pak@kbtu.kz
[1]Жақсылықова Ә.,
докторант, ORCID ID: 0000-0003-0422-7432,
e-mail: a.jaxylykova@kbtu.kz
[3]Комада П.
PhD, профессор, ORCID ID: 0000-0002-9032-9285,
e-mail: p.komada@pollub.pl

[1]Қазақстан-Британ техникалық университеті, Алматы қ., Қазақстан
[2]Ақпараттық және есептеу технологиялары институты, Алматы қ., Қазақстан
[3]Люблин технологиялық университеті, Люблин қ., Польша

НАҚТЫ АУДИТОРИЯ НЕМЕСЕ МАЗМҰНҒА БАЙЛАНЫСТЫ ТЕКСТ ҚҰРАСТЫРУ СТИЛІН АДАПТАЦИЯЛАУ

Аңдатпа

Мәтінді құру стилін белгілі бір аудиторияға немесе мазмұнға бейімдеуге жоғары дәлдіксіз-ақ қол жеткізуге болады. Бұл жұмыста үлгі салмақтарынан бас тартылып, оның орнына: (i) Байес оңтайландыруын қолданып сегіз декодер гиперпараметрі қайталанды; (ii) оқылуды өзгертетін бір жолдық мәнер туралы кеңес қосылды. 8–14B параметрлері бар үш бақылау нүктесі (LLaMA-3.1-8B, DeepSeek-Qwen-8B/14B) және бес математикалық эталон (AQUA-RAT, MathQA, GSM8K, MAWPS, SVAMP) бойынша жүргізілген эксперименттер Optuna-ның 50-сынақтық сәйкестік іздестіру көрсеткіштерін шамамен 3%-ға жақсартқанын көрсетті. 30–70B дәл баптаумен негізгі көрсеткіштермен салыстырғанда 5–10 ұпай айырмашылық байқалды. Сол параметрлер тапсырмалар арасында 2 ұпайдан аз шығынмен қолданылады. Бала аудиториясына бағытталған тақырыпты қосу дәлдікке айтарлықтай әсер етпейді, бірақ Флеш-Кинкейд оқылым ұпайын екі есе төмендетіп, дәлелдеу жолдарын қысқартады. Барлық эксперименттер бір A100 құрылғысында бірнеше GPU сағатында аяқталды, бұл әдісті ресурс шектеулі ортада да тиімді пайдалануға мүмкіндік береді. Зерттеу микробағдарламамен біріктірілген мұқият декодерді басқару қосымша оқыту немесе орнату уақытынсыз сандық дәлдікті және аудиторияға лайықты мәтін ұсынылуын қамтамасыз ететінін көрсетеді.

Тірек сөздер: декодерді оңтайландыру, стильге бейімделу, оқуға жеңілдік, үлкен тілдік модельдер, математикалық сұрақтарға жауап, Байес гиперпараметрін іздеу, Флеш-Кинкейд бағалауы.

[1]*Жанбырбай Ж.,
магистрант, ORCID ID: 0009-0003-5568-6142,
*e-mail: z_zhangbyrbay@kbtu.kz
[2]Ахметов И.,
PhD, профессор, ORCID ID: 0000-0002-3221-9352,
e-mail: i.akhmetov@ipic.kz
[1]Пак А.,
PhD, профессор, ORCID ID: 0000-0002-8685-9355,
e-mail: a.pak@kbtu.kz
[1]Джаксылыкова А.,
докторант, ORCID ID: 0000-0003-0422-7432,
e-mail: a.jaxylykova@kbtu.kz
[3]Комада П.
PhD, профессор, ORCID ID: 0000-0002-9032-9285,
e-mail: p.komada@pollub.pl

[1]Казахстанско-Британский технический университет, г. Алматы, Казахстан
[2]Институт информационных и вычислительных технологий, г. Алматы, Казахстан
[3]Люблинский технологический университет, г. Люблин, Польша

АДАПТАЦИЯ СТИЛЯ СОЗДАНИЯ ТЕКСТА К КОНКРЕТНОЙ АУДИТОРИИ ИЛИ СОДЕРЖАНИЮ

Аннотация

Адаптация стиля генерации текста к конкретной аудитории или содержанию может быть достигнута без дорогостоящей тонкой настройки. Мы отказываемся от модельных весов и вместо этого (i) перебираем восемь гиперпараметров декодера с помощью байесовской оптимизации и (ii) добавляем однострочную стилевую подсказку, которая изменяет удобочитаемость. Эксперименты на пяти математических бенчмарках (AQUA-RAT, MathQA, GSM8K, MAWPS, SVAMP) с тремя контрольными точками с параметрами 8-14 B (LLaMA-3.1-8B, DeepSeek-Qwen-8B/14B) показали, что 50-пробный поиск Optuna повышает точность точного соответствия на 36 процентных пунктов и закрывает 5–10 пунктов разрыва с базовыми точками с точной настройкой 30–70 B. Те же настройки переносятся между задачами с потерей менее двух пунктов. Добавление заголовка, ориентированного на детей, оставляет точность практически неизменной, вдвое снижая уровень оценки по Флешу-Кинкейду и сокращая трассы рассуждений. Все эксперименты укладываются в несколько GPU-часов на одном A100, что делает метод практичным для развертывания в условиях ограниченных ресурсов. Исследование демонстрирует, что тщательный контроль декодера в сочетании с микропрограммами обеспечивает численную корректность и приемлемое для аудитории изложение без дополнительного времени на обучение или настройку.

Ключевые слова: оптимизация декодера, адаптация стиля, читабельность, большие языковые модели, математические ответы на вопросы, байесовский поиск гиперпараметров, оценка Flesch-Kincaid.

Article submission date: 19.05.2025